

## CLAIMS

What is claimed is:

- 1 1. A multi-mode texture decompression method for use during graphics  
2 processing, comprising;  
3 (a) sending a request for compressed texture data to memory;  
4 (b) receiving the compressed texture data from the memory;  
5 (c) identifying at least one of a plurality of compression algorithms associated  
6 with the compressed texture data; and  
7 (d) decompressing the compressed texture data in accordance with the  
8 identified compression algorithm.
- 1 2. The method as recited in claim 1, and further comprising prior to sending  
2 the request, compressing the texture data utilizing the plurality of  
3 compression algorithms, selecting the most favorable compressed texture  
4 data, and storing the most favorable compressed texture data in the  
5 memory.
- 1 3. The method as recited in claim 2, wherein the most favorable compressed  
2 texture data is the most accurate replication of an original version of the  
3 texture data.
- 1 4. The method as recited in claim 2, and further comprising storing a mode  
2 identifier with the compressed texture data.
- 1 5. The method as recited in claim 4, wherein the at least one of the plurality of  
2 compression algorithms associated with the compressed texture data is  
3 identified utilizing the mode identifier.

- 1     6.     The method as recited in claim 5, wherein the mode identifier includes a  
2           mode bit.
- 1     7.     The method as recited in claim 1, wherein at least one of the compression  
2           algorithms represents a 4x4 block of texels of the texture data utilizing two  
3           bits per texel only if the textels are opaque, each 4x4 block of texels  
4           including two 16-bit colors stored in an RGB 565 format and two  
5           additional colors created by interpolating between the two 16-bit colors  
6           stored in the RGB 565 format to form a 4-entry lookup table, where a 2-bit  
7           index is adapted for being used to determine which 16-bit color from the  
8           lookup table is used for each texel of the 4x4 block of texels, and  
9           transparent texels are represented by making one of the four 16-bit colors  
10          transparent.
- 1     8.     The method as recited in claim 1, wherein at least one of the compression  
2           algorithms represents a 4x8 block of texels utilizing three bits per texel,  
3           each 4x8 block of texels including two 15-bit colors stored in an RGB 555  
4           format and five additional colors created by interpolating between the two  
5           15-bit colors stored in the RGB 555 format to form an 8-entry lookup table,  
6           where an eighth 15-bit color is defined to be a transparent color, and a 3-bit  
7           index is used to determine which 15-bit color from the lookup table is used  
8           for each texel in the 4x8 block of texels.
- 1     9.     The method as recited in claim 1, wherein at least one of the compression  
2           algorithms represents a 4x8 block of texels utilizing two bits per texel only  
3           if the textels are opaque, each 4x8 block of texels including four 15-bit  
4           colors in an RGB 555 format to form a 4-entry lookup table, a 2-bit index is  
5           adapted for being used to determine which of the four 15-bit colors is  
6           assigned to each texel.

1 10. The method as recited in claim 1, wherein at least one of the compression  
2 algorithms represents a 4x8 block of texels by two bits per texel, each 4x8  
3 block of texels including three 20-bit colors stored in a 5555 format, where  
4 a first and second one of the 20-bit colors are used for primary colors of a  
5 left 4x4 sub-block of the 4x8 block of texels, and a second and third one of  
6 the colors are used for primary colors of the right 4x4 sub-block of the 4x8  
7 block of texels, where two additional 20-bit colors are created in each 4x4  
8 sub-block of texels by interpolating between the 20-bit colors associated  
9 with the corresponding 4x4 sub-block of texels, where a 2-bit index is  
10 adapted for being used to determine which of the four 20-bit colors is  
11 assigned to each texel, and a lookup table is used to determine which 20-bit  
12 color is applied to each texel.

1 11. A multi-mode texture decompression computer program product for use  
2 during graphics processing, comprising;  
3 (a) computer code for sending a request for compressed texture data to  
4 memory;  
5 (b) computer code for receiving the compressed texture data from the memory;  
6 (c) computer code for identifying at least one of a plurality of compression  
7 algorithms associated with the compressed texture data; and  
8 (d) computer code for decompressing the compressed texture data in  
9 accordance with the identified compression algorithm.

1 12. A multi-mode texture decompression system for use during graphics  
2 processing, comprising;  
3 (a) a texture fetch module adapted for sending a request for compressed texture  
4 data to memory, and receiving the compressed texture data from the  
5 memory;  
6 (b) a format detection module adapted for identifying at least one of a plurality  
7 of compression algorithms associated with the compressed texture data; and

- 8 (c) a plurality of decompression modules coupled between the texture fetch  
9 module and the format detection module, the decompression modules  
10 adapted for decompressing the compressed texture data in accordance with  
11 the compression algorithm identified by the format detection module.  
1
- 1 13. The system as recited in claim 12, wherein at least one of the compression  
2 algorithms represents a 4x4 block of texels of the texture data utilizing two  
3 bits per texel only if the textels are opaque, each 4x4 block of texels  
4 including two 16-bit colors stored in an RGB 565 format and two  
5 additional colors created by interpolating between the two 16-bit colors  
6 stored in the RGB 565 format to form a 4-entry lookup table, where a 2-bit  
7 index is adapted for being used to determine which 16-bit color from the  
8 lookup table is used for each texel of the 4x4 block of texels, and  
9 transparent texels are represented by making one of the four 16-bit colors  
10 transparent.
- 1 14. The system as recited in claim 12, wherein at least one of the compression  
2 algorithms represents a 4x8 block of texels utilizing three bits per texel,  
3 each 4x8 block of texels including two 15-bit colors stored in an RGB 555  
4 format and five additional colors created by interpolating between the two  
5 15-bit colors stored in the RGB 555 format to form an 8-entry lookup table,  
6 where an eighth 15-bit color is defined to be a transparent color, and a 3-bit  
7 index is used to determine which 15-bit color from the lookup table is used  
8 for each texel in the 4x8 block of texels.
- 1 15. The system as recited in claim 12, wherein at least one of the compression  
2 algorithms represents a 4x8 block of texels utilizing two bits per texel only  
3 if the textels are opaque, each 4x8 block of texels including four 15-bit  
4 colors in an RGB 555 format to form a 4-entry lookup table, a 2-bit index is

5 adapted for being used to determine which of the four 15-bit colors is  
6 assigned to each texel.

1 16. The system as recited in claim 12, wherein at least one of the compression  
2 algorithms represents a 4x8 block of texels by two bits per texel, each 4x8  
3 block of texels including three 20-bit colors stored in a 5555 format, where  
4 a first and second one of the 20-bit colors are used for primary colors of a  
5 left 4x4 sub-block of the 4x8 block of texels, and a second and third one of  
6 the colors are used for primary colors of the right 4x4 sub-block of the 4x8  
7 block of texels, where two additional 20-bit colors are created in each 4x4  
8 sub-block of texels by interpolating between the 20-bit colors associated  
9 with the corresponding 4x4 sub-block of texels, where a 2-bit index is  
10 adapted for being used to determine which of the four 20-bit colors is  
11 assigned to each texel, and a lookup table is used to determine which 20-bit  
12 color is applied to each texel.

1 17. A multi-mode texture decompression system for use during graphics  
2 processing, comprising;  
3 (a) means for sending a request for compressed texture data to memory;  
4 (b) means for receiving the compressed texture data from the memory;  
5 (c) means for identifying at least one of a plurality of compression algorithms  
6 associated with the compressed texture data; and  
7 (d) means for decompressing the compressed texture data in accordance with  
8 the identified compression algorithm.

1 18. A multi-mode texture compression method for use during graphics  
2 processing, comprising;  
3 (a) compressing texture data utilizing a plurality of compression algorithms;  
4 (b) selecting the most favorable compressed texture data;  
5 (c) storing the most favorable compressed texture data in memory;

- 6 (d) storing a mode bit with the most favorable compressed texture data in the
- 7 memory;
- 8 (e) sending a request for the compressed texture data to the memory;
- 9 (f) receiving the compressed texture data from the memory;
- 10 (g) determining the mode bit associated with the received compressed texture
- 11 data;
- 12 (h) identifying at least one of the plurality of compression algorithms
- 13 associated with the compressed texture data based on the mode bit; and
- 14 (i) decompressing the compressed texture data in accordance with the
- 15 identified compression algorithm.

- 1 19. A multi-mode texture compression method for use during graphics
- 2 processing, comprising;
- 3 (a) compressing texture data utilizing a plurality of compression algorithms;
- 4 (b) selecting the most favorable compressed texture data;
- 5 (c) storing the most favorable compressed texture data in memory; and
- 6 (d) storing a mode bit with the most favorable compressed texture data in the
- 7 memory;
- 8 (e) wherein the mode bit associated with the received compressed texture data
- 9 is capable of being used to identify at least one of the plurality of
- 10 compression algorithms associated with the compressed texture data such
- 11 that the compressed texture data is capable of being decompressed in
- 12 accordance with the identified compression algorithm.

- 1 20. A data structure stored in memory for compressing texture data
- 2 representing a YxZ block of texels utilizing three bits per texel, each YxZ
- 3 block of texels including two X-bit colors stored in a predetermined format
- 4 and five additional colors created by interpolating between the two X-bit
- 5 colors stored in the Predetermined format to form a lookup table, where an
- 6 eighth X-bit color is defined to be a transparent color, and a W-bit index is

7        used to determine which X-bit color from the lookup table is used for each  
8        texel in the YxZ block of texels.

1    21.    A data structure stored in memory for compressing texture data  
2        representing a YxZ block of texels utilizing two bits per texel only if the  
3        textels are opaque, each YxZ block of texels including four X-bit colors in  
4        a predetermined format to form a lookup table, a W-bit index is adapted for  
5        being used to determine which of the four X-bit colors is assigned to each  
6        texel.

1    22.    A data structure stored in memory for compressing texture data  
2        representing a YxZ block of texels by two bits per texel, each YxZ block of  
3        texels including three X-bit colors stored in a predetermined format, where  
4        a first and second one of the X-bit colors are used for primary colors of a  
5        left YxY sub-block of the YxZ block of texels, and a second and third one  
6        of the colors are used for primary colors of the right YxY sub-block of the  
7        YxZ block of texels, where two additional X-bit colors are created in each  
8        YxY sub-block of texels by interpolating between the X-bit colors  
9        associated with the corresponding YxY sub-block of texels, where a W-bit  
10       index is adapted for being used to determine which of the four X-bit colors  
11       is assigned to each texel, and a lookup table is used to determine which X-  
12       bit color is applied to each texel.